# 원전 안전-필수 소프트웨어 개발 방법론

▌권 기 춘 / 한국원자력연구소


## 요 약

원전 안전등급 계측제어계통이 디지털화 되고 있으나 안전계통을 소프트웨어로 구현하는 과정에는 소프트웨어 공통모드고장(Common Mode Failure)과 같은 위험 요소가 존재한다. 일반 산업체와는 달리 원전에서 발생하는 사고는 일반 공중에게 방사능 누출 가능성을 내재하고 있어 디지털 안전계통은 원자로를 안전하게 정지시키는 것을 보장하고, 방사능이 누출되는 사고에 대해서도 그 피해를 최소화시키거나 방지할 수 있는 기능을 확보해야 한다. 이와 같은 원전 안전계통 디지털화의 성공과 원전의 안전성을 확보하기 위해서는 소프트웨어 신뢰성과 안전성 확보가 관건이며 안전-필수 소프트웨어 검증 기술 개발이 절실히 요구된다. 본 기고에서는 원전 디지털 안전-필수 소프트웨어에 대한 개발방법과 규제요건, 규제방법 및 규제대처방법 등에서 공통적으로 고려해야 할 안전-필수 소프트웨어 개발 및 검증의 기술적 측면과 안전-필수 소프트웨어의 개발 및 검증의 각 단계에 대해서 기술한다.


## 서 론

기존 원자력발전소(원전) 아날로그 계측제어계통은 그 시스템의 노후화와 기술의 낙후성으로 인해 운전 및 유지보수 비용이 증가하고 안전성까지 위협받고 있다. 반면에 컴퓨터 소프트웨어 기술은 급속도로 발전하여 그 성능의 우수성을 인정받고 여러 산업 분야에서 활용되고 있다. 원자력 산업에서도 이러한 추세와 보조를 같이하여 비안전 계측제어계통이 디지털화 되었으며, 소프트웨어 기술의 발전으로 인해 점차 안전계통에 소프트웨어를 사용하고 있는 추세이다. 그러나 고신뢰도를 요구하는 안전계통을 소프트웨어로 구현하는 과정에는 소프트웨어 공통모드고장(Common Mode Failure)과 같은 위험 요소가 존재한다[1]. 일반 산업체와는 달리 원전에서 발생하는 사고의 여파는 일반 공중에게 방사능 누출과 방사성 물질의 피해를 입힐 수 있는 가능성을 내재하고 있어 이러한 가능성을 배제하기 위해 원전 디지털 안전계통은 원자로를 정지시키고 원자로가 안전한 정지조건을 만족하는 것을 보장하고, 원자로 밖으로 방사능이 누출되는 사고에 대해서도 그 피해를 최소화시키거나 방지할 수 있는 기능을 보장하는 것이다[2]. 따라서 이러한 디지털 안전계통의 기능을 보장할 수 있도록 안전-필수 소프트웨어를 개발하기 위해서는 신뢰성(Reliability)과 안전성(Safety) 확보가 그 관건이며 이 두 가지 기능은 원전 계측제어계통 안전-필수 소프트웨어의 특성상 매우 중요하다.

원전 디지털 안전계통에 적용되는 안전-필수 소프트웨어는 엄격한 규제요건을 적용받고 있으며, 이러한 규제요건은 원전의 안전성을 확보하기 위한 필수

적인 요소이다. 국내에서는 한국원자력안전기술원이 원전 건설과 운영에 관한 규제를 시행중이며, 디지털 계측제어 소프트웨어에 관한 규제요건 및 지침을 개발하여 시행중이다. 현재 국내에서도 디지털 안전 소프트웨어 검증은 이와 같은 규제요건에 따라서 수행되고 있다. 안전-필수 소프트웨어의 검증은 소프트웨어 생명주기 전 단계에 걸쳐서 진행되어야 한다.

## 안전-필수 소프트웨어 개발 규제요건

원전 디지털 안전계통은 고신뢰도와 고장시 안전을 보장하고 안전기능의 손실이 없도록 하기 위해서 품질보증 개념과 다양성을 갖는 다중방호(Defense-in-Depth & Diversity) 개념을 가지고 설계되어야 한다. 이러한 개념은 IEEE Std 603-1991, Reg. Guide 1.152[3]와 IEEE 7-4.3.2-1993[4], 미국원자력규제위원회(USNRC)의 Branch Technical Position(BTP)-14, 한국원자력안전기술원의 디지털-기반 계측제어계통에 관한 규제지침등에 잘 나타나 있으며, 규제기관에서 디지털 안전계통을 검토할 때 주요 규제요건으로 사용된다. 그 밖에 많은 Reg. Guide, IEEE 기술기준이 참고로 사용된다. 피규제자는 궁극적으로 IEEE 7-4.3.2를 만족시킬 의무가 있다. IEEE 7-4.3.2는 규제자와 피규제자 사이의 인터페이스, 확인 및 검증(Verification & Validation, V&V)의 독립성, V&V에 사용된 도구와 사람에 대한 자격, 하드웨어와 소프트웨어 및 시스템의 요구사항, 소프트웨어 개발 절차, 상용 소프트웨어의 검증방법, 하드웨어 및 소프트웨어의 통합, 확인 및 검증방법 등을 기술하고 있다. 그러나 IEEE 7-4.3.2에는 시스템 및 하드웨어와의 통합을 고려하면서 주로 소프트웨어에 관련된 규제내용들이 원칙적인 측면에서 간략하게 기술되어 있다. 즉, IEEE 7-4.3.2의 본문내용에는 다른 규제지침서와 표준들로의 참고부분이 많으며 기술적으로 구체적인 내용은 대부분 부록에 수록되어 있다. 따라서 규제기관에서는 많은 부분을 확대 해석하여 적용하고 있으며, 이에 따라 여러 가지 규제지침을 필요로 한다.

안전-필수 소프트웨어 관련 규제지침을 살펴보면, 최상위에 10 CFR(Code of Federal Regulations) Part 50(원자력발전소 설계기준)이 있고 그 밑에 안전계통 설계 기준인 IEEE Std 603이 있으며, 안전성과 관련하여 Reg. Guide 1.153(안전계통 규제요건)[5]과 IEEE Std 379-2000(안전계통 단일고장 기준의 표준)[6]가 있다. 소프트웨어 품질과 관련하여 Reg. Guide 1.152(안전계통 디지털 컴퓨터의 규제요건), Reg. Guide 1.168(안전계통 소프트웨어 확인 및 검증, 검토, 감사 규제요건)[7]이 있으며, 이 규제요건에서 승인한 IEEE Std 7-4.3.2가 있다. 그 하위에 소프트웨어 계획 관련인 IEEE Std 1058.1-1987(소프트웨어 프로젝트 관리계획을 위한 기준)[8], IEEE Std 828-1998(소프트웨어 형상관리계획 작성을 위한 기준)[9], IEEE Std 1042-1987(소프트웨어 형상관리 기준)[10], IEEE Std. 1012-1986(소프트웨어 확인/검증계획을 위한 기준)[11], IEEE Std 1228-1994(소프트웨어 안전성 계획을 위한 기준)[12], Reg. Guide 1.169(안전계통의 디지털 컴퓨터 소프트웨어에 대한 형상관리 계획)[13]이 있다. 그리고 소프트웨어 설계와 관련하여 IEEE Std 830-1993(안전 소프트웨어 요구명세를 위한 기준)[14], IEEE Std 1016-1987(소프트웨어 설계명세 기준)[15], IEEE Std 1016.1-1993(소프트웨어 설계명세 안내서)[16], Reg. Guide 1.172(안전계통 소프트웨어 요구명세 규제지침)[17]이 있으며, 소프트웨어 시험검증과 관련하여 IEEE Std 829-1983( 안전계통의 소프트웨어 시험에 관한 문서화 기준)[18], IEEE Std 1008-1987(안전계통 소프트웨어 단위 시험을 위한 기준)[19], IEEE Std 1059-1993(소프트웨어 시험검증 기준)[20], Reg. Guide 1.171(안전계통의 디지털 소프트웨어 단위시험에 대한 규제 지침)[21]등이 있다. 그 밖에 IEEE Std 730-1998(소프트웨어 품질보증 계획서 작성을 위한 기준)[22], IEEE Std 1074-1997(소프트웨어 생명주기 공정 개발 기준)[23], IEEE Std 1028-1998(소

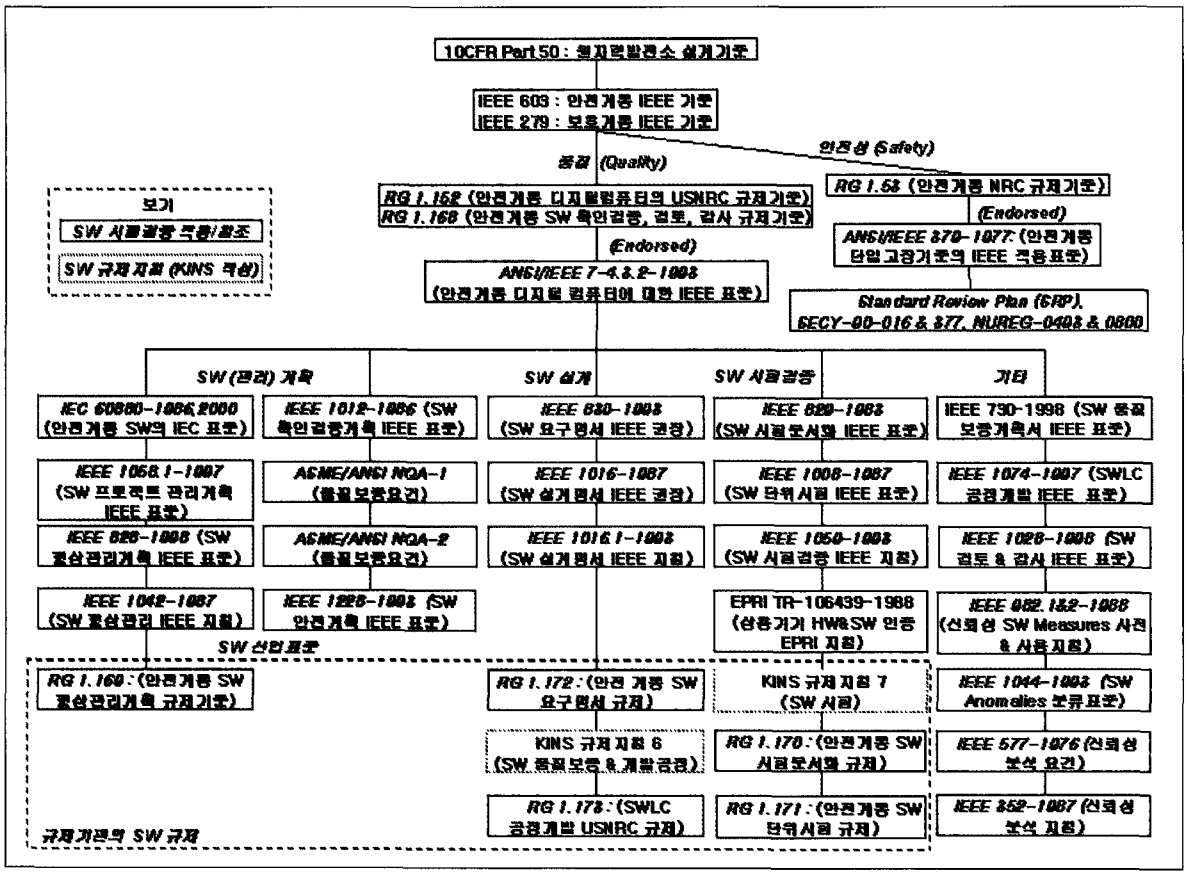



그림 1 안전-필수 소프트웨어 규제요건 계층도

프트웨어 검토 및 감사 기준)[24]이 있다. 이와 같은 규제 기준 또는 지침을 계층적으로 표현하면 [그림 1]과 같다.

## 안전-필수 소프트웨어 개발 및 검증의 기술적 측면

원전 디지털 계측제어 소프트웨어는 그 중요도(Criticality)의 정도에 따라 안전계통 소프트웨어, 제어계통 소프트웨어, 감시계통 소프트웨어와 그 외 여러가지 운전보조계통 등으로 나눌 수 있다. 이들은 다시 안전-필수, 안전-관련과 비안전 소프트웨어로 나누어지며, 이들 소프트웨어를 개발하고 인허가를 수행할 때 현재 쟁점으로 부각되고 있는 것은 대중의 안전과 직결되는 안전-필수 또는 안전-관련 소프트웨어의 품질과 신뢰도의 보장 문제이다.

구체적으로 말하자면 안전계통 소프트웨어의 공통모드고장 극복이 관건이며 이를 위해 소프트웨어 다중성, 주요 소프트웨어 격리, 소프트웨어 위험도 분석, 고장허용(Fault-tolerant) 소프트웨어 개발, 정형기법(Formal Method)의 도입 등 다양한 방법들이 제시되고 있다.

안전-필수 소프트웨어를 개발하고 확인 및 검증을 수행하여 이에 대한 인허가를 획득하기 위해서는 소프트웨어 개발공정 전단계에 걸쳐서 기존의 소프트웨어 공학에서 연구되고 실용화되어 있는 소프트웨어 개발 방법론들이 매우 엄격하게 적용되어야 하며, 이에 대해서 규제기관에서는 [그림 2]와 같이 소

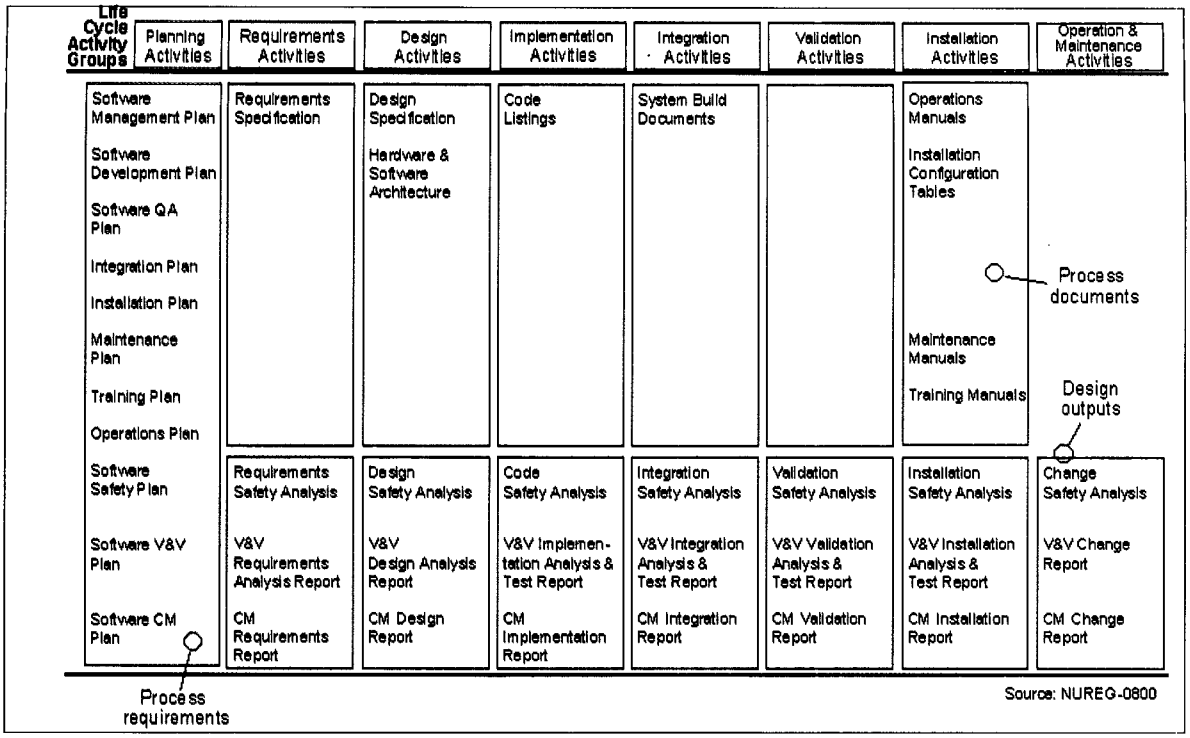



그림 2 소프트웨어 생명주기 업무 정의

프트웨어 생명주기에 대한 업무를 정의하고 있다. 이러한 엄격함에도 불구하고 현재 소프트웨어의 신뢰도 측면에서 소프트웨어 공통모드고장 문제를 완전히 해결하지 못하고 있다.

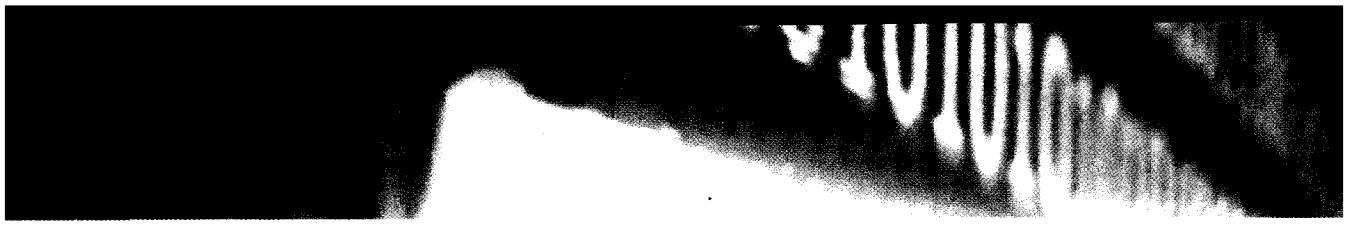
급격한 기술 발전은 소프트웨어 개발자에게 새로운 개발방법의 사용을 요구하며 규제기관에게는 새로운 규제요건의 개발과 새로운 규제방법을 요구하고 있다. 또한 소프트웨어 공학의 학문적인 측면에서도 고품질의 안전-필수 소프트웨어를 적은 비용으로 효율적으로 개발할 수 있고 소프트웨어 공통모드 고장을 극복하여 신뢰도를 보장할 수 있는 이론 정립과 도구 개발에 많은 노력을 하고 있으며, 이를 바탕으로 한 새로운 개발 방법론들을 제시하고 있다.

원전 계측제어 소프트웨어들은 그 중요도의 정도와 소프트웨어의 종류에 따라 분류되며 이러한 분류에 따른 각각의 소프트웨어는 서로 다른 패러다임과 개발 방법론들이 적용되어야 한다. 이를 위해 IEEE Std 7-4.3.2, IEEE Std 1228 등에서와 같이 체계적인 요건들의 개발 및 보완 작업이 진행 중에 있다. 그러나 규제요건 및 표준들이 급격한 기술 발전을 효율적으로 수용하기 위해서는 각각의 요건들이 유연성과 확장성을 갖도록 보완되어야 한다.

## 안전-필수 소프트웨어 개발

### 계획 단계

이 단계에서는 소프트웨어 개발 계획(Development Plan), 관리 계획(Management Plan), 통합 계획(Integration Plan), 보수 계획(Maintenance Plan), 형상관리 계획(Configuration Management Plan), 안전 계획(Safety Plan), 검증 및 확인 계획(Verification and Validation Plan), 품질보증 계획(Quality Assurance Plan) 등의 작성이 수행되며 이와 관련하여 IEEE Std 1058.1, IEEE Std 828, IEEE



Std 730.1-1989[25], IEEE Std 1012, IEEE Std 1228 등의 규제요건들이 있다. 규제기관에서는 소프트웨어 생명주기 전단계에 대하여 IEEE 7-4.3.2를 중심으로 상하의 관련된 기술기준(Codes and Stds)을 참고하여 감사와 검토를 수행하며 이 때 필요한 문서의 종류를 밝히고 있다.

### 요구분석 단계

이 단계에서는 소프트웨어 요구사항 명세서(Software Requirement Specification, SRS)를 작성한다. 요구사항은 기능들의 분할, 기능간의 상관 및 종속관계, 행위들에 의한 기능들의 작동, 분할된 기능 사이의 관계성 등을 기술한다. 이들의 기능은 최상위 수준에서는 외부로부터의 입력과 처리, 외부로의 출력을 기술하며, 하위계층으로 내려가면서 각 기능이 분할된다.

요구분석 단계에서 수학적인 정형기법을 도입하여 품질과 신뢰도를 높여 보고자 하는 것이 최근의 추세이다. 요구사항을 형식에 맞게 작성함으로써 다음 단계인 설계를 수행할 때 보다 완벽하게 요구사항을 반영할 수 있고 요구분석을 수동 혹은 자동으로 수학적인 증명을 할 수 있다. 그리고 오류를 개발공정 초기에 발견 수정할 수 있음으로써 개발비용을 절감할 수 있는 장점이 있다. 그러나 이 단계의 주목적이 사용자 혹은 시스템 엔지니어와 소프트웨어 엔지니어사이의 의견교환을 통해서 정확한 요구사항을 수립하는 것인데, 이러한 수학적인 정형기법의 도입은 자연 언어를 사용한 통신보다는 의사 전달이 부자유스러운 단점이 있다. 이러한 정형화된 요구조건으로부터 테스트 케이스를 자동 생성할 수 있고 요구사항 자체를 실행할 수 있는 자동화 도구를 만들어 요구분석 단계에서 개발하고자 하는 시스템의 동작을 모의화해 봄으로써 오류를 조기에 발견, 해결하고자 하는 노력을 진행 중에 있다.

### 설계 단계

설계단계에서는 소프트웨어 설계 명세서(Software Design Specification, SDS)를 작성한다. 여기서도 앞 단계에서와 같은 수학적인 정형기법의 도입이 시도되고 있으며 설계 방법론에서도 이제까지의 구조적 설계 방법론에서 점차 객체지향 설계 방법론으로 바뀌어 가고 있다. 모듈 결합도(Module Coupling), 모듈강도(Module Cohesion), 모듈화(Modularity), 정보은닉(Information Hiding)등 그동안의 소프트웨어 공학에서의 연구 결과들이 점차 객체지향 설계 방법론으로 초점이 맞춰지고 있다. 이러한 객체지향 설계방법은 재사용성을 높여 개발하는 소프트웨어 시스템의 신뢰도를 높이고 V&V 비용을 감소시킬 수 있다.

### 구현 단계

개발시 사용되는 프로그래밍 언어가 점차 어셈블리 언어에서 C, C++ 및 Ada와 같은 고수준 언어 또는 PLC 언어인 Ladder logic이나 Functional Block Diagram(FBD)로 바뀌어 가고 있다. 이러한 프로그래밍 언어의 선택과 소프트웨어 공통모드고장 극복을 위한 다양성을 언어와 컴파일러에 적용하고 있다. 또한 컴파일러, 운영체제, 시험 도구, 에디터등 개발에 사용되는 모든 도구들에 대한 사전 자격심사(Prequalification)가 이루어지고 있다. 소프트웨어 개발 회사들은 자체적으로 코딩 지침서등을 작성하여 사용하고 있으나 표준화가 되어 있지 못한 형편이다. 완벽한 예외사항 처리등 고장허용 프로그래밍이 요구되고 있으며 고장허용 소프트웨어의 개발은 요구분석 단계부터 고려되어야 할 기술적인 사항이 매우 많다. 이를 위해 현재 시도되고 있는 기술들 중 대표적인 것으로 소프트웨어 생명주기 전 단계에 걸쳐 다양성을 고려하는 N-version 프로그래밍 기법과 Recovery Block 기법이 사용되고 있으나 소프트웨어 복잡도 증가에 따른 안전성 위해요소가 있어 많은 논란이 제기되고 있다.

### 통합단계

요구분석 단계, 설계단계, 구현단계를 거치면서

세부 모듈의 프로그래밍이 되고 나면, 이러한 Break-down의 역순으로 모듈들을 통합하고 하드웨어와 통합되어 시스템을 완성하고 통합 순서에 따라 확인과 시험이 이루어진다. 이러한 시험은 모듈시험, 통합시험, 기능시험, 시스템 시험, 인수시험, 설치시험의 순서로 이루어진다. 요구되는 신뢰도에 따라 시험계획 작성과 시험종료 시점이 달라지며 안전-필수 소프트웨어일수록 시험비용이 올라간다. 근본적으로 소프트웨어의 완벽한 시험은 불가능하며 확인 및 검증의 독립성, 자체시험, 확인 및 검증의 심도등이 주 관심사이다. 현재 사용중인 시험관련 규제지침에는 IEEE Std 829와 IEEE Std 1008이 있으며 확인 및 검증과 관련하여 IEEE Std 1012가 있다. 원전 안전계통 소프트웨어와 같은 안전-필수 소프트웨어의 시험을 위해서는 이러한 기존의 표준을 바탕으로 엄격하고 명확하며 객관적인 기준이 제시되어야 하며, Reg. Guide 1.171에서 이러한 기준의 일부를 명시하고 있다.

#### 설치, 운영 및 유지보수 단계

이 단계에서는 설치문서, 운전 및 유지보수 지침서, 훈련지침서등 각종 문서들을 정확히 갖추어야 하며 개발과정에서 발견하지 못한 결함을 보완하기 위한 유지 및 보수계획, 회귀분석(Regressing) 시험계획과 이에 따른 형상관리 수정, V&V 이상 보고서, 안전성 분석 수정 등이 수행된다. 특히 소프트웨어 유지보수는 계측제어계통이 하드웨어에서 소프트웨어로 바뀜에 따라 현장에서 느끼는 가장 심각한 애로 사항으로 부각되고 있어 소프트웨어 시스템에 대한 시험가능성 여부는 새로운 문제로 제기되고 있다.

## 안전-필수 소프트웨어 검증

#### 요구분석 단계 검증

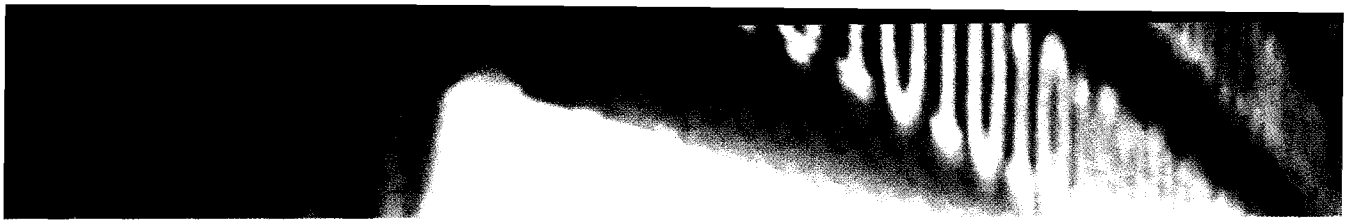
이 단계의 검증은 소프트웨어 요구사항명세서에 명세된 요구사항, 즉 소프트웨어의 기능 및 동작이 시스템 기능명세서 등에 기술된 요건을 BTP-14에서 요구하는 기능특성과 공정특성으로 나누어 만족한다는 것을 체계적으로 보여주는 것이다. 검증방법으로 인허가 적합성 검토, 소프트웨어 요구사항에 대한 시스템 요구사항을 양방향으로 추적하는 추적성 분석, Fagan 인스펙션, 그리고 Model checking, Theorem proving 또 는 Graphical back animation/Simulation 등의 정형 기법이 있다. 그리고 요구분석 단계를 포함하여 모든 단계에서 적용되는 형상관리 평가가 검증 업무에 포함된다.

#### 설계단계 검증

근본적으로 설계단계의 검증은 요구분석 단계 검증과 크게 다르지는 않으나 그 대상이 소프트웨어 설계명세서라는 것이다. 이 단계에서는 BTP-14에서 제안하는 승인기준에 따라서 개발한 점검항목에 따라 검토하는 인허가 적합성 검토, 소프트웨어 품질을 검증하기 위한 검토 방법중의 하나인 Fagan 인스펙션, 모델 체킹(예, New Symbolic Model Verifier, Model Checker/Certifier)이나 정리 증명(예, Prototyping Verification System)등의 정형 기법을 적용하여 소프트웨어 설계명세에 대한 철저한 검증을 수행한다. 그리고 설계단계에서 단위시험 계획 및 시험설계 업무를 수행한다.

#### 구현단계 검증

구현단계에서는 개발된 소프트웨어 코드에 대해서 BTP-14에서 제시하는 검토기준을 근간으로 작성된 검증 체크 리스트를 활용하여 이를 기반으로 추적성 분석, 인터페이스 분석, 소프트웨어 구조 분석, 상세 검증 체크 리스트를 이용한 상세설계 검증을 수행한다. 그리고 설계단계에서 작성된 단위시험 계획 및 시험설계를 사용하여 시험사례(Test Case) 및 시험절차(S/W Test Procedure)를 생성하고, 그리고 단위시험(Unit Testing)을 실행하여 단위시험보고서를 작성한다.



### 통합단계 검증

통합단계 검증에서는 통합단계에서 이루어지는 각종 소프트웨어 모듈들을 하나의 프로그램으로 통합하고, 통합된 프로그램을 하드웨어 및 계기와 서로 통합하여 최종적으로 통합된 결과물을 시험하는 과정에 대해서 검증하는 것이다. 통합 코드에 대한 인스펙션과 통합 테스트 및 계통 테스트가 수행된다. 각각의 테스트단계에서 테스트 계획, 테스트 설계, 테스트 케이스 및 테스트 절차의 생성과 테스트 수행과 테스트 보고서 생성과 같은 6가지 테스트 활동을 수행해야 한다.

### 안전성 분석

원전 계측제어계통의 디지털화 추세에 따라 컴퓨터 소프트웨어의 안전성 확보가 중요한 현안으로 부각되고 있다. 이에 따라 인허가 기관이나 산업표준 기구 등에서 안전성 심사 및 개발기준이 되는 인허가 요건과 국제표준 정립 노력을 기울이고 있다. 안전성 분석은 안전-필수 소프트웨어의 검증에서 다른 비안전 소프트웨어 검증과 비교하여 특별히 차이가 나는 부분이다. 안전-필수 소프트웨어를 개발하기 위해서는 계획단계에서 안전계획을 작성해야 하며 이에 따라 안전성 분석을 수행하여야 한다. 소프트웨어 안전계획은 시스템 안전현안이 소프트웨어 개발 중에 적절히 고려되도록 하는 기본 문서이다. 만약 소프트웨어 안전계획이 없다면 안전현안들이 적절히 고려되지 않거나, 또는 해결되지 않을 수 있다.

소프트웨어 안전성 분석 기술에는 체크 리스트나 HAZOP (Hazard and Operability)에 의한 방법 이외에도 고장모드영향분석(Failure Mode Effect Analysis, FMEA), 고장수목기법(Fault Tree Analysis, FTA)와 같은 기법이 사용될 수 있다. 원전 계측제어계통 소프트웨어 요구분석 단계에서는 안전성 분석을 수행하기 위해 HAZOP 방안을 활용한다. HAZOP은 화학공장과 같은 산업에서 시스템 안전성 분석을 위해 사용하던 기법이다. 설계단계에서도 체크 리스트와 HAZOP에 의한 안전성 분석을 수행한다. 구현단계에서는 구현된 코드에 대해서 상세 안전성 분석을 위하여 고장수목기법을 사용한다.

## 결 론

지금까지 원전 디지털 계측제어계통에서 사용되는 안전-필수 소프트웨어 규제요건과 안전-필수 소프트웨어 개발과 검증측면에서 문제점을 논하였다. 이러한 기술적 현안을 해결하기 위한 연구 노력들을 살펴보았고, 안전-필수 소프트웨어 규제요건이 갖추어야 할 기준을 서술하였다. 이와 같은 규제요건이 가져야 할 기준은 곧 안전-필수 소프트웨어 개발방법과 규제방법의 기준이 될 수 있을 정도로 직접적인 관계가 있다.

안전-필수 소프트웨어의 품질보증 관련 현안들을 소프트웨어 개발단계별로 알아보았고 각 단계별 기술 추세를 소프트웨어 개발방법, 규제방법, 규제기준의 측면에서 제시하였다. 그리고 소프트웨어 공통모드고장 대책과 같은 안전성 보장 관련 현안에 관하여 논하였다.

후속기 원전에서 계측제어계통을 디지털화하고 기술자립을 추구하고 있는 우리도 이러한 상황에서 안전-필수 소프트웨어 안전성 보장 기술과 확인 및 검증 기술과 같은 핵심 기반 기술의 확보가 필수적이라 하겠다.

### [참고문헌]


[1] 이장수외, "원전 계측제어 고신뢰도 소프트웨어 확인/검증의 기술 현황," 원자력학회 논문지, Vol. 26, No.4, pp.600~610, 한국원자력학회, December 1994.

[2] IEEE Std 603-1991, IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations.

[3] Regulatory Guide 1.152, "Criteria for Digital Computers in Safety Systems of Nuclear Power Plants," Rev. 1, U.S. Nuclear Regulatory Commission, January 1996.
[4] IEEE Std 7-4.3.2-1993, "IEEE Standard for Digital Computers in Safety Systems of Nuclear Power Generating Stations."
[5] Regulatory Guide 1.153, "Criteria for Power, Instrumentation, and Control Portions of Safety Systems," Rev. 1, U.S. Nuclear Regulatory Commission, 1996.
[6] IEEE Std 379-2000, "IEEE Std. Application of the Single-Failure Criterion to Nuclear Power Generating Station Safety Systems."
[7] Regulatory Guide 1.168, "Verification, Validation, Reviews and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," U.S. Nuclear Regulatory Commission, 1997.
[8] IEEE Std 1058.1-1987, "Standard for Software Project Management Plans."
[9] IEEE Std 828-1983, "Standard for Software Configuration Management Plans."
[10] IEEE Std 1042-1987, "Guide to Software Configuration Management."
[11] IEEE Std 1012-1992, "Standard for Software Verification and Validation Plans."
[12] IEEE Std 1228-1994, "Standard for Software Safety Plans."
[13] Regulatory Guide 1.169, "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," U.S. Nuclear Regulatory Commission, 1997.
[14] IEEE Std 830-1984, "Guide for Software Requirements Specifications."
[15] IEEE Std 1016-1987, "Recommended Practice for Software Design Descriptions."
[16] IEEE Std 1016.1-1993, "Guide to Software Design Descriptions."
[17] Regulatory Guide 1.172," Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," U.S. Nuclear Regulatory Commission, 1997.
[18] IEEE Std 829-1983, "Standard for Software Test Documentation."
[19] IEEE Std 1008-1987, "Standard for Software Unit Testing."
[20] IEEE Std 1059-1993, "IEEE Guide for Software Verification and Validation."
[21] Regulatory Guide 1.171, "Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," U.S. Nuclear Regulatory Commission, 1997.
[22] IEEE Std 730-1990, "IEEE Standard for Software Quality Assurance Plans."
[23] IEEE Std 1074-1995, "IEEE Standard for Developing Software Life Cycle Processes."
[24] IEEE Std 1028-1988, "Standard for Software Reviews and Audits."
[25] IEEE Std 730.1-1989, "Standard for Software Quality Assurance Plans."
[26] NUREG-0800, "Standard Review Plan," U.S. Nuclear Regulatory Commission, June 1997.